TUTORIAL

Q. Given the relations PROFESSOR (PID, PNAME, DEPT, SALARY) and STUDENT (ROLLNO, NAME, CLASS, ADVISER) where ADVISER is a foreign key to PROFESSOR, write one equivalent relational algebra expression for each of the following:

i. $\Pi_{PNAME}(\sigma_{SALARY>30000}(PROFESSOR))$

ii) $\Pi_{SNAME}(\sigma_{DEPT='CSE'}(STUDENT \bowtie_{ADVISER=PID} PROFESSOR))$

→

i) SALARY = $\sigma_{SALARY>30000}(PROFESSOR)$

RESULT = $\Pi_{PNAME}(SALARY)$

ii) DEPT_CSE = $\sigma_{DEPT='CSE'}$

ADVI

= STUD

~~DEPT_CSE = $\sigma_{DEPT='CSE'}(PROFESSOR)$~~

~~= DEPT_CSE $\bowtie_{ADVISER=PID}$~~

ii) DEPT_CSE = $\sigma_{DEPT='CSE'}(PROFESSOR)$

STUDENT_CSE = STUDENT $\bowtie_{ADVISER=PID}$ (DEPT_CSE)

RESULT = Π_{SNAME} (STUDENT_CSE)

Q2. Consider the following schema:

Suppliers (sid, sname, address)

Parts (pid, pname, color)

catalog (sid, pid, cost)

The key fields are underlined. Foreign key in Catalog are sid and pid refering to Suppliers and Parts respectively. Write the relational algebra expression for the following queries

i) Find the sids of suppliers who supply some red or green part

ii) Find the sids of suppliers who supply every part.

iii) Find the names of suppliers who supply some red part

→ i)

RED_GREEN_PARTS = $\sigma_{<color='Red'> \text{ OR } <color='Green'>}$ (Parts)

PART_SUPPLIERS = Catalog $\bowtie_{pid=pid}$ (RED_GREEN_PARTS)

RESULT = π_{sid} <PART_SUPPLIERS>

ii) REQ_PARTS = $\pi_{<sid, pid>}$ (Catalog $\bowtie_{pid=pid}$ Part)

RESULT 1 = $\pi_{<pid>}$ (Parts)

RESULT = REQ_PARTS $\div$ RESULT 1

iii)

RED_PART = $\sigma_{<color='Red'>}$ (Parts)

RED_SUPPLIERS = Catalog $\bowtie_{pid=pid}$ (RED_PART)

SUPPLIERS_DETAILS = Suppliers $\bowtie_{sid=sid}$ (RED_SUPPLIERS)

RESULT = π_{sname} (SUPPLIERS_DETAILS)

Q3. Consider two tables STUDENT (ROLLNO, NAME, CLASS) and ENROLLMENT (ROLLNO, COURSENAME) where ROLLNO in ENROLLMENT is a foreign key referring to STUDENT. It is required that every time a STUDENT tuple is deleted, all ENROLLMENT tuples referring to the deleted STUDENT tuple are also deleted. Write SQL statements to specify this foreign key requirement.

→
```
CREATE TABLE ENROLLMENT(
    ROLLNO  INTEGER
    COURSENAME  VARCHAR(15)  NOT NULL,
    PRIMARY KEY (ROLLNO, COURSENAME)
    FOREIGN KEY (ROLLNO) REFERRENCES STUDENT (ROLLNO),
                                                    );
        ON DELETE CASCADE;
```

Q.41 Consider the relational database shown below:

employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)

Define the integrity constraints in the given relation. State the assumptions used.

→

- In employee table, person-name is the primary key hence it doesn't have any ~~primary~~ duplicate or null values. person-name is a foreign key in company relation
- ~~Primary~~ Primary key in works relation is person-name and company-name as it has unique values. person-name and company-name is also foreign key with references to employee relation and company relation respectively.
- Primary key of company is company-name.
- manages relation, person-name + manager-name is the primary key. Here foreign key is person-name which references to employee.

Q5) Consider the foll. relations for a database that keeps track of business trips of salesperson in a sales office.

SALESPERSON (Ssn, Name, StartYear, DeptNo)
TRIP (Ssn, FromCity, ToCity, DepatureDate, ReturnDate, TripId)
EXPENSE (TripId, AccountNo, Amount)

a) A trip can be charged to one or more account. Specify the foreign keys for this schema, stating any assumptions you make.

b) TRAVEL_CITY = $\sigma_{\langle FromCity = 'Mumbai' \rangle AND \langle ToCity = 'Delhi' \rangle}$ (TRIP)

PERSONS = TRAVEL_CITY $\bowtie_{ssn=Ssn}$ (SALESPERSON)

PERSON_EXPENSES = PERSONS $\bowtie_{Tripid=Tripid}$ (EXPENSE)

PERSON-AMNT = $\sigma_{\langle AMOUNT > 30000 \rangle}$ (PERSON_EXPENSES)

RESULT = $\pi_{\langle ssn, Sname, StartYear, DeptNo \rangle}$ (PERSON-AMNT);

c) ~~Assume, a tab relation ACCOUNT which has attributes AccountNo, AcntName~~

a) A ~~trip~~ can be charged to more than one account similarly an account can have various trip ids. This is a M:N relationship.

Assume, a relation ACCOUNT which has attribute AccountNo, AccountName.

ACCOUNT (AccountNo, AccountName)

where AccountNo is a primary key,

Hence AccountNo in EXPENSE relation will be foreign key.